(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 January 2001 (25.01.2001)

PCT

(10) International Publication Number
WO 01/06385 A1

(51) International Patent Classification[7]: G06F 15/16

(21) International Application Number: PCT/US00/18791

(22) International Filing Date: 10 July 2000 (10.07.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/144,118 16 July 1999 (16.07.1999) US

(71) Applicant *(for all designated States except US)*: NETCONVERGENCE, INC. [US/US]; 3335 Kifer Road, Santa Clara, CA 95051 (US).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: FOK, Simon, K. [US/US]; 3335 Kifer Road, Santa Clara, CA 95051 (US).

(74) Agent: CHAN, H., C.; NetConvergence, Inc., 3335 Kifer Road, Santa Clara, CA 95051 (US).

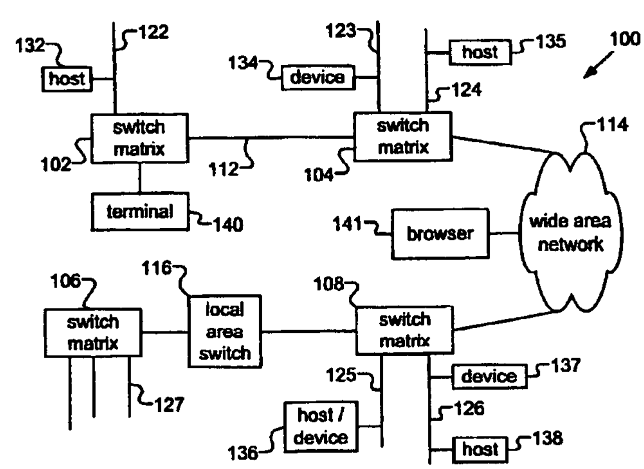
(81) Designated States *(national)*: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*
— *Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: A COMMUNICATION SYSTEM FOR GENERAL CONNECTION INTERFACE MACHINES



(57) Abstract: The present invention is a data communication system (100) that allows communication between general machines (132, 134, 135, 136, 137, 138) having different protocols. A switch matrix (102, 104, 106, 108) contains at least a general connection interface port that can support one or more general machines. These machines and the port communicate using a defined protocol. Another interface port can be installed in the same switch matrix or a different switch matrix that is far away from the first switch matrix. The second port can support one or more machines that communicate using a different protocol. The machines connected to the first port and the second port may be any combination of hosts and/or devices (132, 134, 135, 136, 137, 138). A machine connected to the first port can send commands to, or receive command from, a specific machine connected to the second port, even though they normally cannot communicate with one another because they use different protocols.

WO 01/06385 A1

A COMMUNICATION SYSTEM FOR GENERAL CONNECTION INTERFACE MACHINES

Background of the Invention

The present invention relates to communication system and general connection interface machines. and more specifically, a system that allows general connection interface machines. each operating under a different protocol, to communicate with each other over a wide area.

Computer networking technology has grown tremendously during the past decade. This leads to faster communicate speed between connected machines and wider geographic areas to deploy the machines. As a result of this new connected environment, many new communication architectures and applications have been developed. One example is the popularity of the World-Wide-Web, in which millions of servers and client computers are connected using the so-called "TCP/IP" protocol. Another example is the evolution of connection topology between storage resources. Recently, a new connection topology, called the storage area network ("SAN"), allows any-to-any connection between servers and storage devices. Under this topology, any server connected to a SAN can use any storage device connected to the SAN.

One of the limitations of the conventional networked resource architectures is that the connected machines need to speak the same protocol. For example, under current SAN implementation, Fibre Channel is the connection protocol. However, there are many legacy storage devices that use the Small Computer System Interface ("SCSI") protocol. It will be difficult to replace all these legacy storage devices with new Fibre Channel storage devices. Consequently, there is a need to develop an architecture that allows networking of machines of different protocols.

Summary of the Invention

The present invention is a wide-area data communication system that allows communication between general connection interface machines having different protocols. One embodiment of the present invention involves a switch matrix having at least a general connection interface port that can support one or more general connection interface machines. These machines and the port communicate using a

predefined protocol. Another general connection interface port can be installed in the same switch matrix or a different switch matrix that is far away from the first switch matrix. The second general connection interface port can support one or more general connection interface machines that communicate using a different protocol. The machines connected to the first port and the second port may be a host (i.e., capable of sending commands), a device (i.e., capable or responding to commands), or a combination of hosts and devices. One aspect of the present invention is that a machine connected to the first port can send commands to, or receive command from, a specific machine connected to the second port, even though they normally cannot communicate with one another because they use different protocols.

In this embodiment, each port is associated with a virtual general connection interface module. A physical machine (labeled the first machine) connected to this port communicates with the virtual module. An address mapping function is used to map the module with another physical machine (labeled the second machine) attached to another port (either in the same or a different switch matrix). Similarly, the second machine is associated with a virtual module that is in turn associated with the first machine. Each switch matrix has a switch module. Through the mapping function and the switch module, information can be delivered from the virtual module associated with the first machine to the virtual module associated with the second machine. The virtual module then delivers the information to its associated physical machine. As a result, the physical machines in different ports can communicate with each other.

In order for the switch matrixes to communicate with one another over long distances, each switch matrix contains an interswitch interface. The interswitch interface can be connected to a local or wide area network. Alternatively, the interswitch interfaces of two switch matrixes can be connected directly.

In another embodiment of the present invention, each switch matrix contains a management module. A user can access the management module in a switch matrix (via a user terminal or the Internet) and issue management commands to the switch matrix or other switch matrixes. If a management command is intended for another switch matrix, the switch module of the user's switch matrix delivers the command to its interswitch interface. The command is then delivered to the interswitch interface of the other switch matrix. The switch module inside that matrix will deliver the command to its management module, which then executes the command.

These and other features and advantages of the present invention are described by the following detailed description of the embodiments together with the accompanying drawings.

Brief Description of the Drawings

Fig. 1 is a schematic diagram of a wide-area data communication system of the present invention.

Fig. 2 is a block diagram of a switch matrix of the present invention.

Fig. 3 is a block diagram of a virtual general connection interface module of the present invention.

Fig. 4A is a block diagram of a management module of the present invention.

Fig. 4B is a flow chart showing the operation of a startup module of the present invention.

Figs. 5A and 5B are schematic diagrams showing the address mapping of the present invention.

Fig. 6 shows the data format of a packet of the present invention.

Fig. 7 is a block diagram of an exemplary switch fabric of the present invention.

Detail Description of the Present Invention

The present invention comprises a novel system and related methods for a wide-area communication system. The following description is presented to enable any person skilled in the art to make and use the invention. Description of specific applications is provided only as examples. Various modifications to the preferred embodiments will be readily apparent to those skilled in the art, and the general principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the invention. Thus, the present invention is not intended to be limited to the embodiments shown, but is to be accorded the widest scope consistent with the principles and features disclosed herein.

Fig. 1 shows a schematic diagram of a wide-area data communication system 100 of the present invention. System 100 comprises a plurality of switch matrixes (such as matrixes 102, 104, 106 and 108). These matrixes can communicate with each other through various communication means. For example, matrix 102 can communicate with matrix 104 through a direct link 112; matrix 104 can communicate

with matrix 108 through a wide area network 114; and matrix 108 can communicate with matrix 106 through a local area network 116. Each matrix can manage a plurality of general connection interface ("GCI") machines. In the present invention, GCI refers to an interface system for connecting machines that communicate and transfer information using a predefined protocol. Some examples of popular GCI are Small Computer System Interface (SCSI), Fibre Channel, Firewire and Universal Serial Bus (USB). In Fig. 1, switch matrix 102 is connected to a connection 122 and switch matrix 104 is connected to connections 123 and 124. Other connections 125-127 are also shown in Fig. 1. It should be noted that a connection may comprise of two or more connectors (e.g., each connector may be a copper wire). For example, a parallel-type connection may have more than eight connectors, while a simple serial-type connection may need only two connectors.

A GCI machine 132, generally known as a host, is attached to connection 122 and a GCI machine 134, generally known as a device, is attached to connection 123. In the present invention, a host is a machine in a GCI system that initiates GCI commands, and a device is a machine in a GCI system that services GCI commands. An example of a host is a networked thin-client computer and an example of a device is a disk drive. Examples of GCI commands are "read" and "write" commands. Note that some GCI machines can function as both a host and a device. An example of such a machine 136 is a network computer containing a shared disk drive. One aspect of the present invention is the ability to handle these kinds of machines efficiently.

In the present invention, each GCI machine can communicate with other GCI machines served locally by a switch matrix. For example, a host 135, connected to a connection 124 of switch matrix 104, can communicate with a device 134 connected to connection 123 served locally by switch matrix 104. At the same time, each GCI machine can also communicate with other GCI machines connected to remote switch matrixes. For example, host 132 can send commands to a device 137 served by matrix 108, and device 137 can return data to host 132. One aspect of the present invention is a mapping protocol that allows global communication by all the GCI machines.

The switch matrixes of the present invention can be managed by users. In Fig. 1, a user can connect a terminal 140 to one of the switch matrixes (such as matrix 102), and the user can issue management commands to any switch matrix in system 100. Alternatively, the user can use a separate computer running a regular browser

application 141 to send commands to the switch matrixes (such as matrixes 104 and 108) in system 100 through wide-area network 114.

In the present invention, management command travels through the same communication paths used by regular commands and data of the GCI machines (e.g., direct link 112, wide area network 114 and local area network 116). This communication method is called "inband management." Further details of this method will be below.

Description of Switch Matrixes

Fig. 2 shows a block diagram of a switch matrix 200 of the present invention. Matrix 200 can be used as any of the matrixes 102, 104, 106 and 108 of Fig. 1. Matrix 200 contains a switch module 202 that performs the switching functions of the present invention. It also contains a plurality of GCI ports, such as ports 204 and 206. Each port can be connected to a GCI connection, such as connections 210 and 212. These ports contain the necessary hardware and software for communicating with GCI machines connected to the corresponding connections. For illustrative purposes, connection 210 is a SCSI bus and connection 212 is a USB bus. Each connection can be connected to a plurality of hosts, devices, and host-device enabled machines. Fig. 2 shows a representative GCI machine 207 connected to connection 210 and a representative GCI machine 208 connected to connection 212.

In matrix 200, a virtual GCI module is installed to handle machines connected to each GCI port. For example, virtual GCI modules 214 and 216 are installed to handle GCI machines connected to GCI ports 204 and 206, respectively. The virtual GCI module is a software module that sends and receives commands and data to and from GCI machines. Fig. 3 shows the structure of a virtual GCI module. Each GCI module contains a virtual host module 254 (for interacting with a physical device) and a virtual device module 256 (for interacting with a physical host). Virtual host module 254 is able to send commands to a physical device and virtual device module 256 is able to respond to commands from a physical host. As an example, a virtual SCSI module serves as virtual GCI module 214 of switch matrix 200 because physical GCI machine 207 connected to port 202 is a SCSI machine. If GCI machine 207 is a SCSI device, virtual host module 254 can send a "read" command to GCI machine 207. On

the other hand, if GCI machine 207 is a SCSI host, virtual device module 256 can respond to a "read" command from machine 207.

Matrix 200 also contains a management module 218 for managing the operations of matrix 200. Module 218 is connected to a user interface module 220. This module interacts with a user terminal 222. It allows a user to enter commands to control the operation of switch matrix 200 and other matrixes in communication system 100. Matrix 200 is connected to user terminal 222 directly or via a network connection. Alternatively, user interface module 220 can interact with a browser through wide area network 114 of Fig. 1.

Management module 218 allows a user to configure the switch matrixes in communication system 100. The configuration information (e.g., the addresses of the hosts and devices attached to each matrix) is stored in a nonvolatile memory 224. This information is accessible by switch module 202, and is used for delivering information to appropriate machines. Management module 218 may also communicate with various virtual GCI modules.

Matrix 200 optionally contains an interswitch interface 230 for connecting matrix 200 to other matrixes through a link 231. Link 231 may be connected to a network (e.g., local or wide-area). Thus, link 231 may be connected to local area switch 116 or wide area network 114 of Fig. 1. Using interswitch interface 230, GCI machines 207 and 208 can communicate with a large number of other GCI machines located far from matrix 200. Matrix 200 contains a switch address 226 so that it can be uniquely identified in communication system 100.

In matrix 200, the virtual GCI modules 214 and 216, management module 218 and interswitch interface 230 are connected to switch module 202 via switch interfaces 232-235, respectively. These switch interfaces provides a uniform protocol for individual modules to communicate with switch module 202.

Management Module

Management module 218 performs two basic functions: (a) setting up configuration of a local switch matrix (i.e., the matrix in which module 218 resides) and (b) setting up a global linkage so that all the GCI machines in communication system 100 can communicate with each other.

Management module 218 allows a user to enter configuration information using user terminal 222. For example, the user can assign (or change) switch address 226. A user can also issue commands to management module 218 (e.g., requesting it to scan the ports to determine the types of machines attached to the switch matrix).

Fig. 4A shows the structure of software components used in management module 218 for setting up local configuration. It contains a startup module 272 that executes at system startup. One of its functions is to determine the structure of switch matrix 200 (e.g., the types and IDs of machines attached to each GCI port). The information is preferably stored in a configuration table inside nonvolatile memory 224. For those machines that can handle hot plug-and-play (i.e., a USB-type GCI machine that can be attached or removed from a connection without the need to re-boot the system), management module 218 also contains a heart-beat inquiry module 274. It periodically sends out heart-beat inquiries to the plug-and-play machines so that information about the switch module can be kept current.

Fig. 4B is a flow chart 280 of the operation of startup module 272. In step 282, module 272 reads a nonvolatile system configuration file (not shown) to determine the number and types of GCI ports that have been installed in the switch matrix. In step 284, module 272 loads the appropriate virtual GCI modules matched to the ports. For example, if a SCSI port is installed in the switch matrix, a SCSI-compatible GCI module is loaded to match with the SCSI port. Similarly, if a USB port is installed in the matrix, a USB-compatible GCI module is loaded to match with the USB port. In step 286, startup module 272 requests each GCI module to send inquiries to individual machines attached to the port to determine their characteristics (e.g., device ID. etc.). In step 288, the information obtained by startup module 272 is stored in a status table inside memory 224.

As mentioned above, the management module can also be used to set up a global linkage so that all the GCI machines in communication system 100 can communicate with other GCI machines. A user can connect a terminal to any switch matrix, and use the management module therein to set up the linkage. Each switch matrix is identifiable using its switch address. More details of this aspect of the invention will be described below.

Address Mapping

If a host is connected to a GCI port, a user can select devices in system 100 that are accessible by this host. In the present invention, the device can be located anywhere in communication system 100. The number of devices may be limited by the associated GCI protocol. This is because some GCI protocols place an upper limit on the number of devices that can be connected to a host. For example, a SCSI host can communicate with up to fifteen SCSI devices.

For simplicity, communication between GCI machines connected to a single switch matrix is first described. As explained in more detail below, in order extend the mapping across multiple switch matrixes, the matrix address needs to be included in the mapping.

Every GCI machine attached to a switch matrix has an address (either host address or device address). The complete address of a machine in the switch matrix consists of (a) the machine's ID on its own connection, based on its GCI protocol (the GCI ID), and (b) the port number of the port on the switch matrix to which that connection is attached to. As an example, under SCSI protocol, each machine has a "LUN", which corresponds to the machine's ID in this disclosure. The machine is normally connected to a SCSI port. This port may be one of several SCSI ports (e.g., some SCSI cards contain two ports each). Each port is assigned a port number. Thus, the combination of the port number and the LUN uniquely identify the machine attached to the switch matrix.

The addressing mechanism of a GCI device is first described. In the system of the present invention, a real GCI device (connected to a first port) is mapped to a virtual device in a second port (implemented as a virtual device module loaded into a the second port). For example, a real GCI device 302 in Fig. 5A is mapped into a virtual device 304 (implemented by loading a virtual device module 305 in the switch matrix). Virtual device 304 then emulates the real device to a host 306 that is connected to the switch matrix. Virtual device 304 serves as a conduit for forwarding GCI commands to, and receiving responses from, real device 302. Note that the real GCI device need not follow the same GCI protocol as the host. For example, real GCI device 302 may be a USB device while real host 306 may be a SCSI host. Also, a single real GCI device can be mapped to multiple virtual devices (each implemented by loading a corresponding virtual device module into a GCI port in the switch matrix).

Note that the mapping of a GCI device preferably allows a transformation of the device's GCI ID so that a user can assign any valid ID to virtual device 304 with respect to the second port. This assigned ID may be different from the actual ID of real device 302 in the first port.

To complete the communication link, a real host is mapped as a virtual host in a similar manner as the above-described device. For example, real host 306 can be mapped into a virtual host 308. A virtual host module 309 is loaded in the switch matrix to implement virtual host 308.

The above-described addressing mechanism is implemented using address mapping functions, described in detail below, that take a partially specified address and produce a fully specified address. The full address is used by switch module 202 to route GCI commands and data from one GCI machine to another. In an embodiment of the present invention, switch matrix commands "sendGCICmd()" and "replyGCICmd()" use the output of the mapping function to setup the transfer of the GCI information across switch module 202.

As mentioned above, there are GCI machines that can function as a host and a device. In this case, the host and device are mapped individually using the same method disclosed above.

Fig. 5B shows the mapping functions used in the present invention. The mapping function **D** is the target mapping function. **D** takes the device ID assigned to virtual device 304 (DID*) as inputs and produces a full address of real device 302. This address is then used by switch module 202 to route information to the correct port (in this case, port 1) and to select the correct device attached to that port (e.g., LUN=3). In this example, the mapping function **D** resides in virtual device module 305. As disclosed above, a particular real device can be mapped to multiple virtual devices simultaneously. In this case, each of the multiple virtual device modules associated with the multiple virtual devices has a different mapping function **D** because a different input DID* is used in each module's mapping function to produce the same real device address.

The mapping function **H** is the host mapping function. Function **H** takes the virtual host ID (HID*) and produces the full address of the real host ID. Each virtual host module has its own host mapping function. The same real host can be mapped to different virtual hosts. **H** may also have a default host mapping sentinel value, which is

used when no host mapping has taken place. When the virtual host module receives this value it uses its own host ID default on the connection.

Another mapping function **R** is the return address function. **R** takes the real host ID and produces the return address for the real host. This mapping need not be user configurable. The return address is used to route the command's response back to the originating host.

As an example, device 302 in Fig. 5A has been mapped to host 306's connection through virtual device module 305. The real device ID is DID, but this device shows up as DID* on the connection to which it has been mapped. Host 306 has been mapped from the actual ID, HID, to HID* in the port connecting to device 302. Thus, if host 306 issues a command to device 302 it must select DID*. Virtual device module 305 responds to this DID* using its mapping function **D** and produces a command packet that is sent across the switch module to virtual host 308. In this embodiment, all commands are sent by calling sendGCICmd() with the outputs of the mapping functions as its inputs. A reply to a command is sent using replyGCICmd() using only the command structure as its input.

The command sendGCICmd() calls the appropriate switch module transport functions and parses the mapping results as required. The command replyGCICmd() calls the appropriate switch module transport function and parses the mapping results information embedded into the command packet as required. The information carried in the packet originally sent by sendGCICmd() should be sufficient to get the command packet back to the host without any information provided by sources outside of the transport function.

As mentioned above, the mapping description provided so far is applicable for communication between GCI machines connected to a single switch matrix. In order to provide a complete address across the whole communication system 100, the matrix address of the switch matrix is included as an additional parameter in the mapping. As an example, the matrix address may be the Internet Protocol (IP) address. As a result, each GCI machine can be uniquely identified by its (a) ID in a connection (e.g., the LUN), (b) the port number of the port that handles the connection, and (c) the matrix address of the switch matrix (e.g., the IP address). As an example, the mapping function **D** now takes the form:

D(DID*) => (matrix address, port number, device ID).

Switch Interface

As mentioned above, management module 218, virtual GCI modules 214 and 216, and interswitch interface 230 are attached to switch interfaces 234, 232, 233 and 235, respectively. These switch interfaces convert the information from the original modules to a format understandable by switch module 202. Fig. 6 shows a data format of a packet 320 that can be used for this purpose. Packet 320 contains a header portion, a payload portion, and a checksum (error detection) portion. The header and checksum portions are used by switch module 202 for switching purposes. On the other hand, the payload portion contains digital data generated by virtual GCI modules 214 and 216 or management module 218. For example, the payload may contain a GCI command, GCI data (originated from a virtual GCI module), or configuration information (originated from management module 218).

The checksum portion contains a checksum (calculated using conventional methods) of the digital data in the header and payload portions. The header portion preferably contains the following fields: (a) a "magic" value used for consistency checking, (b) a "source address," i.e., the mapped address of the sender, (c) a "destination address," i.e., the mapped address of the receiver of the packet, (d) a "type" for indicating the consumer of the packet, such as switch module 202, management module 218, interswitch interface 230, or a virtual GCI module 214 or 216, (e) a "sequence number" of the packet, (f) a "packet number" designating the total number of packets that contain the complete information, and (g) a "length" for indicating the byte count of the present packet. These fields allow switch module 202 to process the information efficiently.

Switch Module

Switch module 202 could be a conventional switch fabric that delivers packets from their source to their destination. An exemplary switch fabric 350 is shown in Fig. 7. It comprises a bank 352 of input buffers (labeled buffers IN-1, IN-2 and IN-N) and a bank 354 of output buffers (labeled buffers OUT-1, OUT-2 and OUT-N). The input and output buffers are linked by an interconnection fabric 356. The operation of fabric 356 is controlled by an arbiter 358. Each input buffer is associated with a selected one of the virtual GCI modules, management module or interswitch interface. For example,

GCI modules 214 and 216 are associated with buffers IN-1 and buffer IN-2, respectively. In this case, data from GCI modules 214 and 216 is sent to buffer IN-1 and buffer IN-2, respectively, through connections 362 and 363, respectively. Similarly, each output buffer is associated with a selected one of the virtual GCI modules, management module or interswitch interface. For example, GCI modules 214 and 216 are associated with buffers OUT-1 and buffer OUT-2, respectively. In this case, data intended for GCI modules 214 and 216 is routed to buffer OUT-1 and buffer OUT-2, respectively, which is then delivered to the respective GCI modules through connections 366 and 367.

As mentioned above, data from GCI modules 214 and 216 may have different protocols because the connections they are associated with interface using different protocols. Consequently, interconnection fabric 356 also contains a protocol translator 360. As a result, the data delivered to output buffers OUT-1, OUT-2, ... , OUT-N conforms to their respective output protocol.

InBand Management

As mentioned above, the management module in any of the switch matrixes can be used to manage communication system 100. Data and command packets generated by the management module of a first switch matrix and intended for a second switch matrix are routed by the switch module of the first switch matrix to its interswitch interface. The packets are then delivered to the second switch matrix and received by its interswitch interface. When the switch module of the second switch matrix realizes that the packets are intended for its management module (e.g., by reading the "type" field of the header), the switch module will route the packets to its management module. The management module of the second switch matrix then executes the command or uses the received data. As a result, the management module of the first switch matrix can manage the second switch matrix.

The invention has been described with reference to specific exemplary embodiments thereof. Various modification and changes may be made thereunto without departing from the broad spirit and scope of the invention. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense; the invention is limited only by the provided claims.

//

CLAIMS

I claim:

1. A switching apparatus having at least a first and a second general connection interface port, said first general connection interface port being connected to at least a first machine that can send a command to said first general connection interface port using a first protocol, said first machine having a first identification number known to said first general connection interface port, said second general connection interface port being connected to at least a second machine that can respond to a command from said second general connection interface port using a second protocol, said second machine having a first identification number known to said second general connection interface port, said apparatus comprising:

a first virtual module, coupled to said first general connection interface port, that can respond to a command from said first machine, said first virtual module having a second identification number known to said first general connection interface port;

a second virtual module, coupled to said second general connection interface port, that can send said command to said second machine, said second virtual module having a second identification number known to said second general connection interface port;

an address mapping function for mapping said first machine to said second virtual module and said second machine to said first virtual module; and

a switch module for delivering information from said first virtual module to said second virtual module using said mapping function;

whereas said command is sent by said first machine to second machine through said first virtual module, said switch module and said second virtual module.

2. The apparatus of claim 1 further comprising a first switch interface coupling between said first virtual module and said switch module and a second switch interface coupling between said second virtual module and said switch module.

3. The apparatus of claim 2 wherein said switch interface further comprises means for forming a packet that contains said command.

4. A communication system connected to at least a first machine that can send a command using a first protocol and a second machine that can respond to a command using a second protocol, said first machine having a first identification number known to said first protocol, said second general connection interface port being connected to at least, said second machine having a first identification number known to said second protocol, said system comprising:

a first switch apparatus; and

a second switch appratus;

said first switch apparatus comprising:

a first general connection interface port connected to said first machine,

a first virtual module, coupled to said first general connection interface port, that can respond to a command from said first machine, said first virtual module having a second identification number known to said first general connection interface port;

a first address mapping function for mapping said second machine to said first virtual module; and

a first switch module for delivering information from said first virtual module to said second switch apparatus using said first mapping function;

said second switch apparatus comprising:

a second general connection interface port connected to said second machine,

a second virtual module, coupled to said second general connection interface port, that can send said command to said second machine, said second virtual module having a second identification number known to said second general connection interface port;

a second address mapping function for mapping said first machine to said second virtual module; and

a second switch module for delivering information from said first switch apparatus to said second virtual module using said second mapping function;

whereas said command is sent by said first machine to second machine through said first and said second switch apparatus.

5. The apparatus of claim 4 further comprising a first switch interface coupling between said first virtual module and said first switch module and a second switch interface coupling between said second virtual module and said second switch module.

6. The apparatus of claim 5 wherein each of said first and said second switch interfaces further comprise means for forming a packet that contains said command.

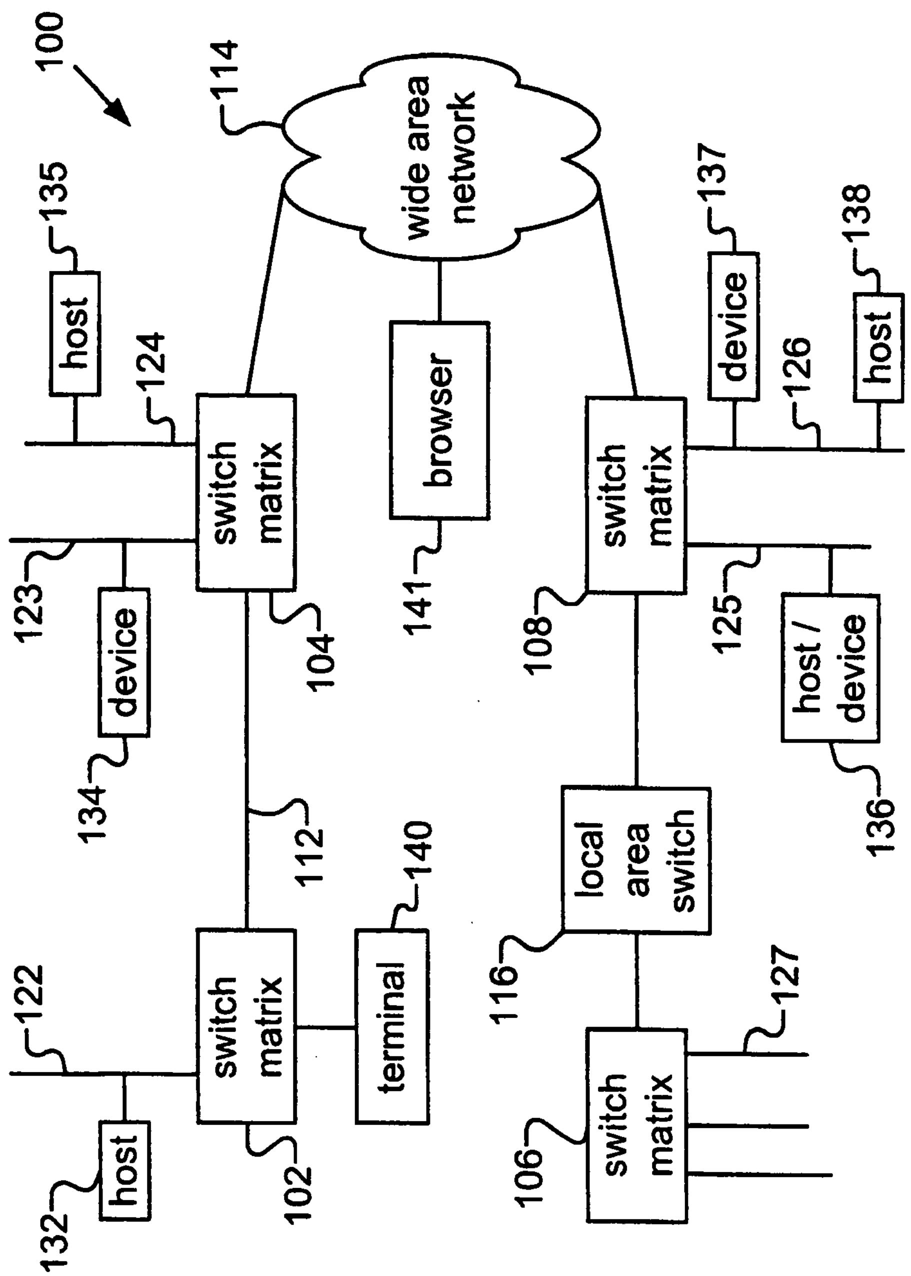
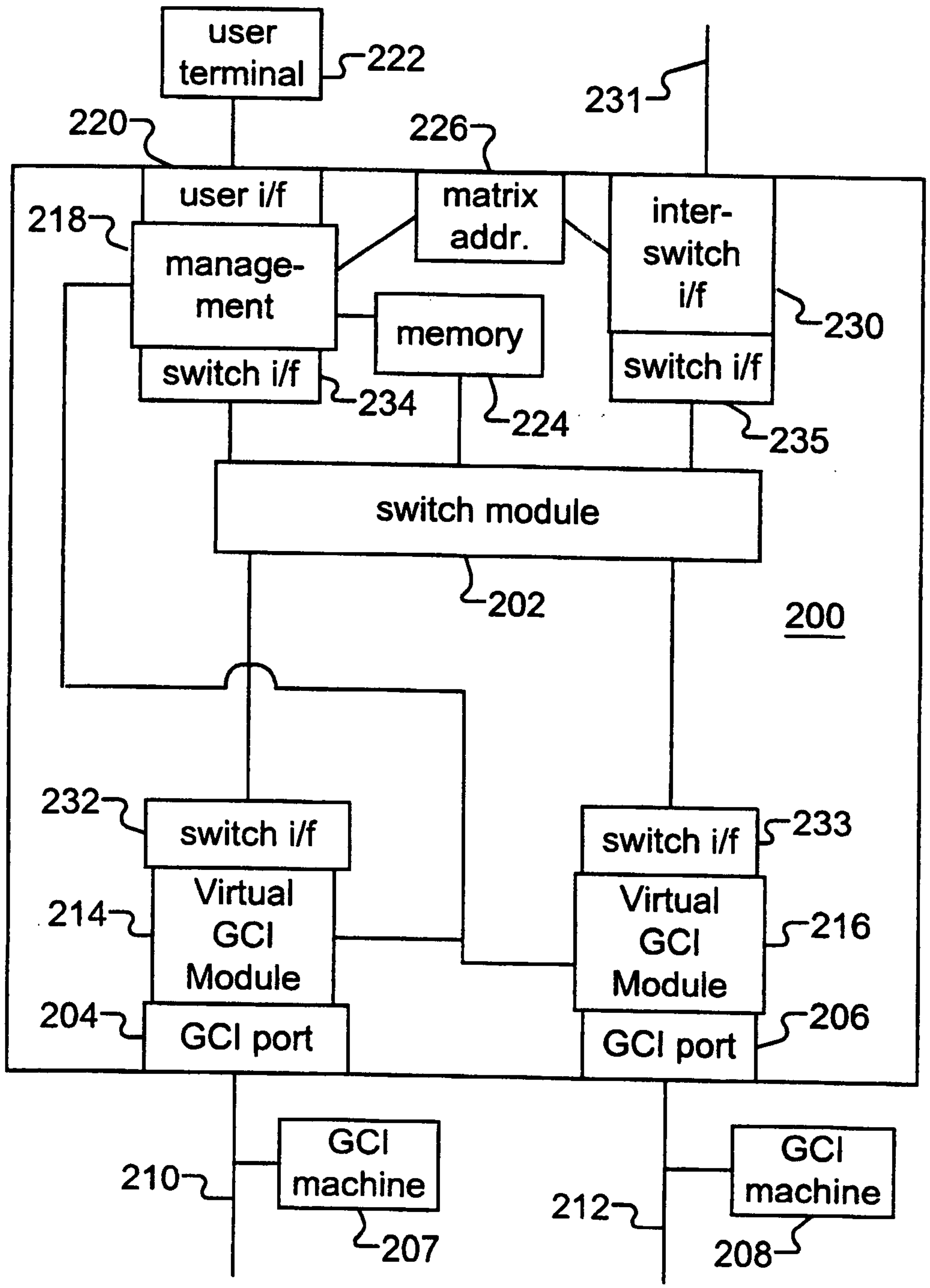

100
host
135
wide area network
114
124
device
137
host
138
switch matrix
browser
switch matrix
126
123
device
104
141
108
125
host / device
134
112
140
local area switch
136
122
switch matrix
terminal
116
127
host
102
switch matrix
106
132

Fig. 1



user terminal
222
220
226
231
user i/f
matrix addr.
inter-switch i/f
218
manage-ment
memory
230
switch i/f
switch i/f
234
224
235
switch module
202
200
232
switch i/f
switch i/f
233
214
Virtual GCI Module
Virtual GCI Module
216
204
GCI port
GCI port
206
GCI machine
GCI machine
210
212
207
208

Fig. 2

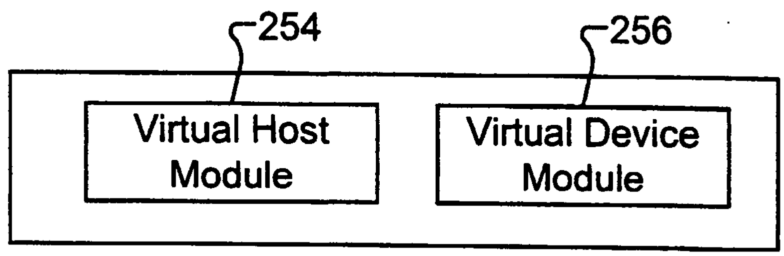


Fig. 3

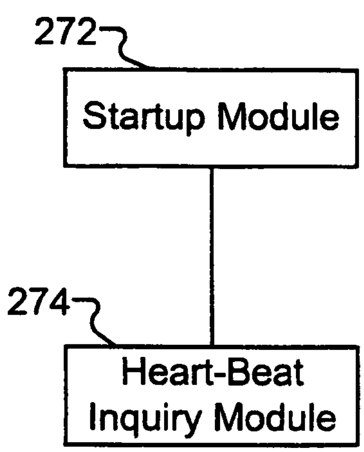


Fig. 4A

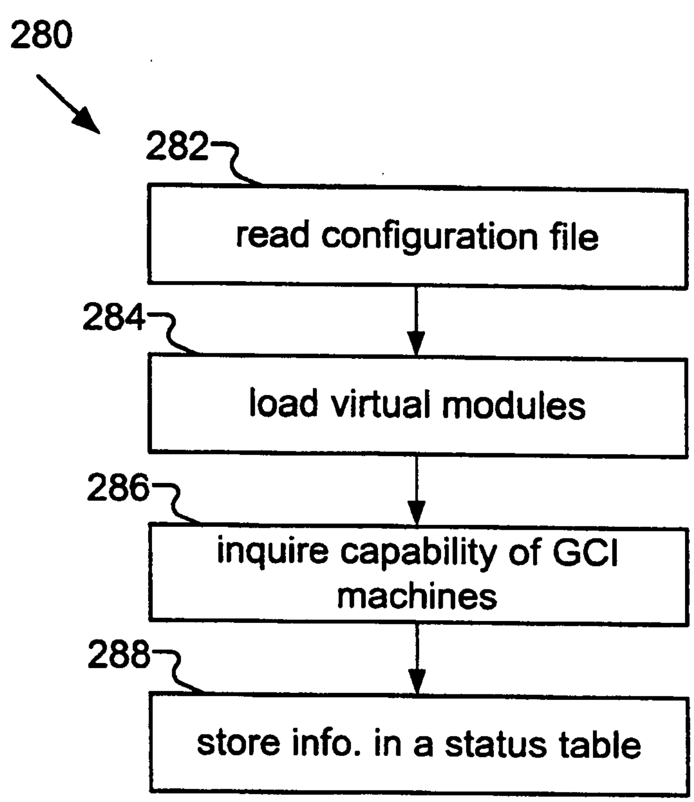


Fig. 4B

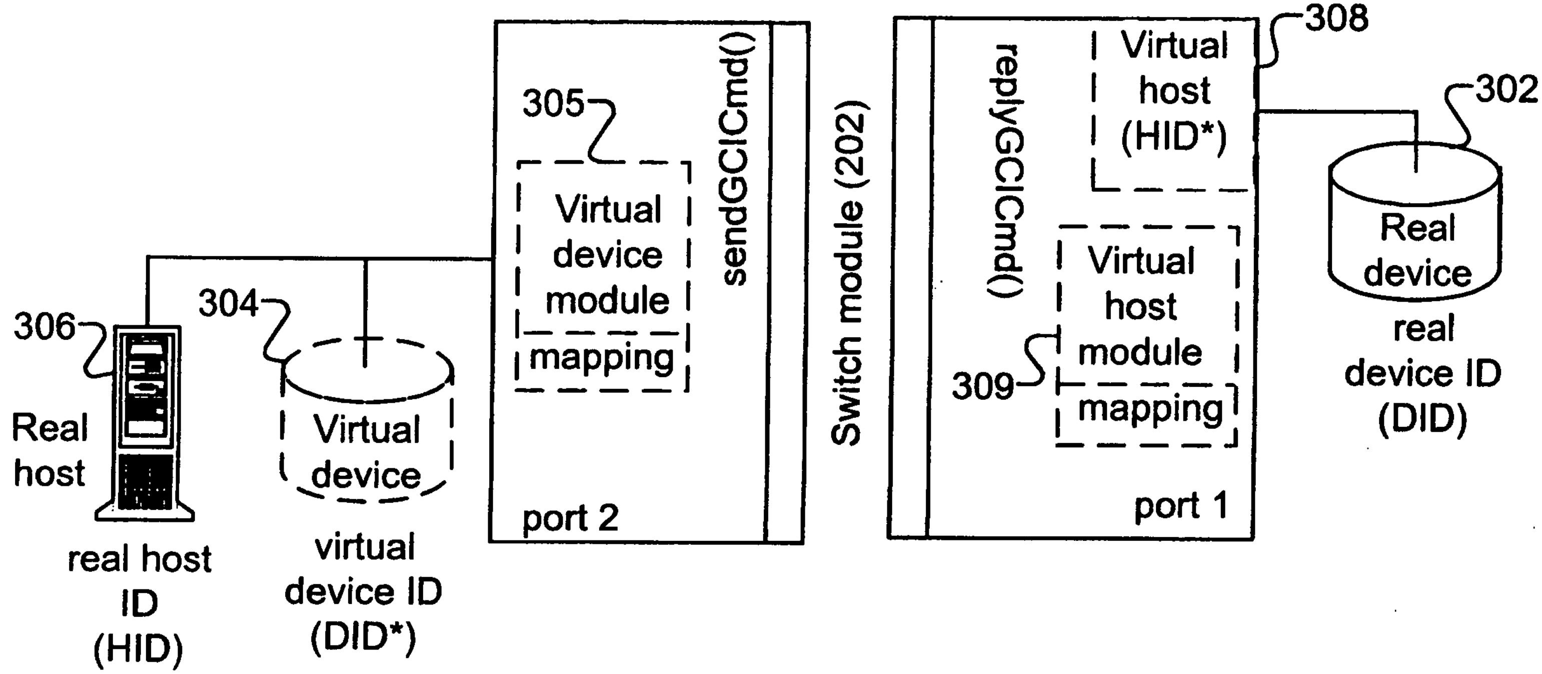


Fig. 5A

D (DID*) $\Longrightarrow$ (port number, DID); R(HID) $\Longrightarrow$ (port number, HID)

H (HID*) $\Longrightarrow$ (port number, HID)

Fig. 5B

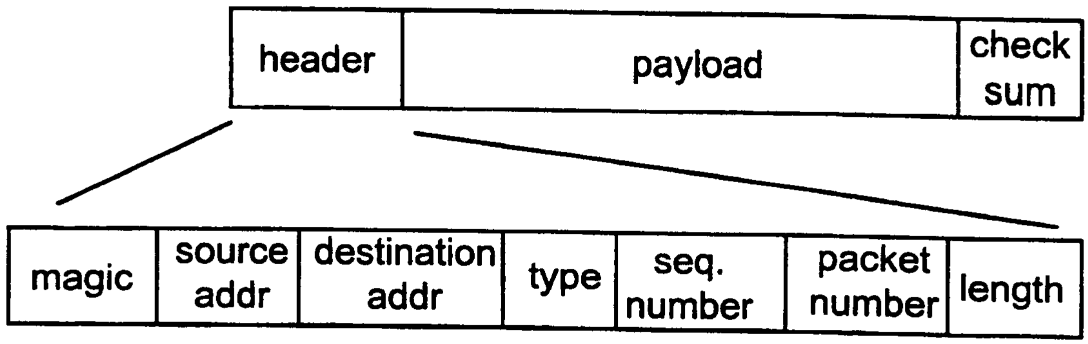

header
payload
check sum
magic
source addr
destination addr
type
seq. number
packet number
length

Fig. 6

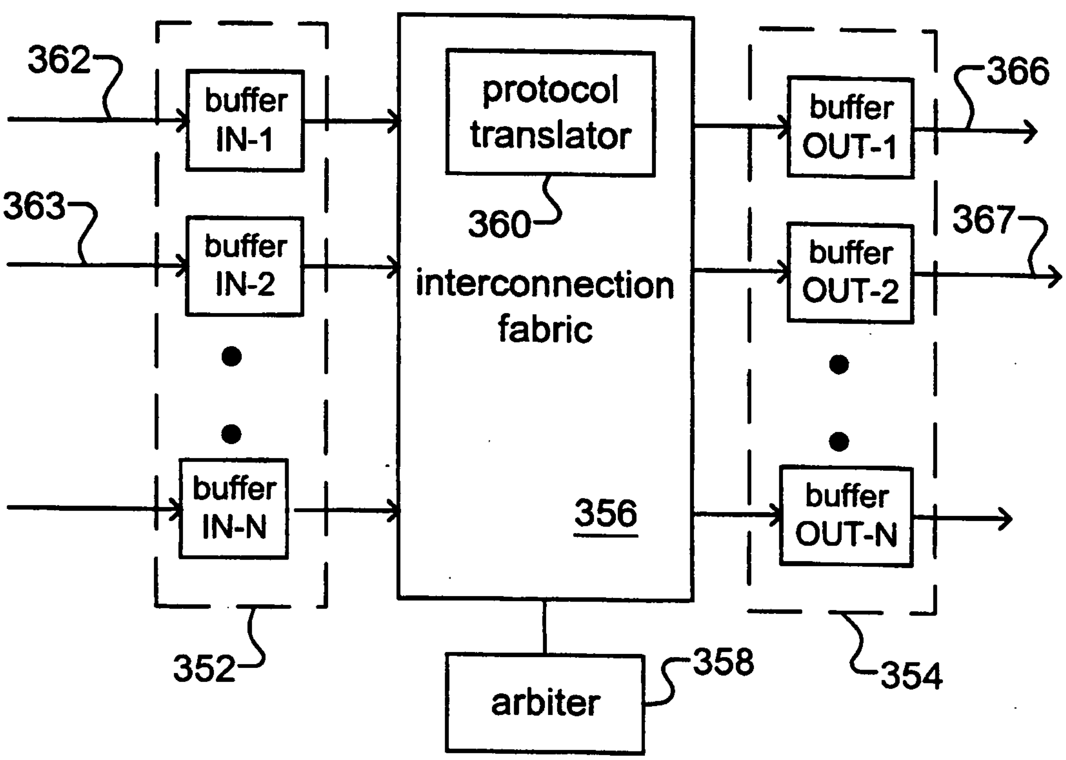

362
363
buffer IN-1
buffer IN-2
buffer IN-N
protocol translator
360
interconnection fabric
356
buffer OUT-1
buffer OUT-2
buffer OUT-N
366
367
352
arbiter
358
354

Fig. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/18791

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :G06F 15/16
US CL :Please See Extra Sheet.
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 709/227, 228, 230, 245, 249; 370/354, 466, 467

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

Please See Extra Sheet.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,371,852 A (ATTANASIO et al.) 06 DECEMBER 1994 Abstract, Figures 2, 4, 5A, 5B, 10, Column 4, Line 42 through Column 5, Line 56, Column 7, Line 38 through Column 8, Line 2, Column 8, Lines 25-58, Column 9, Line 63 through Column 11, Line 20, Column 16, Line 53 through Column 19, Line 3	1-6
Y	US 5,621,734 A (MANN et al.) 15 APRIL 1997 Abstract, Figures 1, 3A, 5, 6A, 6B, Column 3, Lines 12-40, Column 6, Line 62 through Column 8, Line 23, Column 10, Lines 6-63, Column 11, Lines 4-63	1-6
A	US 5,856,974 A (GERVAIS et al.) 05 JANUARY 1999 Entire document	1-6

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
31 OCTOBER 2000	27 NOV 2000

Name and mailing address of the ISA/US	Authorized officer
Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	MARC THOMPSON
Facsimile No. (703) 305-3230	Telephone No. (703) 305-3900

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/18791

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,867,662 A (RIGGS et al.) 02 FEBRUARY 1999 Entire document	1-6
A	US 5,918,019 A (VALENCIA) 29 JUNE 1999 Entire document	1-6
A, P	US 6,006,264 A (COBLY et al.) 21 DECEMBER 1999 Entire document	1-6

A. CLASSIFICATION OF SUBJECT MATTER:
US CL :

709/249

B. FIELDS SEARCHED
Electronic data bases consulted (Name of data base and where practicable terms used):

EAST patent text search. IEEE conference/journal text search
Terms: distributed gateway, protocol translation, WAN/LAN/MAN, data/computer session(s), switch matrix, port, virtual connection, address mapping